

Japanese Patent Laid-open No. HEI10-333902 A

Publication date : December 18, 1998

Applicant : N I I C JOHO SYST:KK

Title : COMPUTER SYSTEM WITH TAMPER CHECKING FUNCTION

5

(57) [Abstract]

[Problems to be Solved] At boot-up of a computer, it is checked whether or not an OS, an OS loader, and other files to be checked are tampered, and after confirmation of their safety, the OS, the OS loader and the application are started.

10 [Solving Means] A boot program 2 stored in a ROM 1 reads first stored information 41 stored in an auxiliary storage 4 not via an OS after initialization of a BIOS, and reads, in accordance with read contents, a file to be checked 42 without involving the OS. These read contents and second stored information stored in the ROM 1 are used to check presence or absence of the file to be checked. After confirming that there has
15 occurred no tamper, an OS loader 43 is called to start up an OS 44.

[Scope of Claims for Patent]

[Claim 1] A computer system having a tamper checking function comprising in which a boot program in a ROM starts up at a boot-up, the boot program starts an OS loader in
20 an auxiliary storage, and the OS loader starts an OS in the auxiliary storage, wherein the boot program comprises:
a stored information reading unit that reads a first stored information stored in the auxiliary storage;
a file to be checked reading unit that reads an arbitrary number of files to be
25 checked from the auxiliary storage based on a description read in the first stored

information; and

a tamper detecting unit that checks presence or absence of tamper in each of the files to be checked based on second stored information, the first stored information which is read and each of the files to be checked in the boot program.

5 [Claim 2] A computer system, wherein

a digital signature for each of files to be checked is stored in the first stored information,

a key for verifying the signature is stored in the second stored information, and

the tamper detecting unit according to claim 1 checks presence or absence of
10 tamper of each of the files to be checked from the digital signature and the key for verifying the signature.

[Claim 3] A computer system, wherein an OS loader stored in the auxiliary storage is, as one of the files to be checked, subjected to tamper check at boot-up.

[Claim 4] A computer system, wherein an OS file stored in the auxiliary storage is, as
15 one of the files to be checked, subjected to tamper check at boot-up.

[Claim 5] A computer system, wherein

the first stored information includes information for tamper check of the first stored information,

the tamper detecting unit according to claim 1 has a function of checking
20 tamper of the first stored information.

[Claim 6] A computer system, wherein

a compressed value which is obtained by compressing each of the files to be checked by a one-way function and a digital signature of the first stored information are stored in the first stored information,

25 a key for verifying the signature is stored in the second stored information, and

the tamper detecting unit checks presence or absence of tamper of the first stored information by using the digital signature and the key for verifying the signature, and checks presence or absence of tamper of each of the files to be checked by using the compressed value.

5 [Claim 7] A computer system, wherein

the boot program comprises a stored information obtaining unit that obtains the second stored information from an input device, and

the second stored information is obtained by inputting of a user.

[Claim 8] The computer system according to claim 7, wherein

10 a compressed value obtained by compressing each of the files to be checked and the second stored information by a one-way function, and a compressed value obtained by compressing the first stored information and the second stored information by a one-way function are stored in the first stored information,

the tamper detecting unit calculates the compressed value again by using the
15 second stored information obtained by the stored information obtaining unit according to claim 7 and compares a resultant value with the compressed value stored in the first stored information to check presence or absence of tamper of each of the files to be checked and the first stored information.

[Claim 9] A computer system, wherein

20 the second stored information is included in an external storage,

the boot program has a second stored information reading unit that obtains the second stored information from the external storage, and

the external storage comprises an anti-tamper unit of the second stored information.

25 [Claim 10] A computer system in which the second stored information is stored in the

external computer, wherein

the external computer comprises:

an anti-tamper unit of the second stored information; and

a tamper detection calculation unit that carries out calculations necessary for

5 tamper detection using the second stored information,

the boot program comprises communicating unit with the external computer,

the tamper detecting unit according to claim 1 checks presence or absence of

tamper of each of files to be checked by using the first stored information, each of the

files to be checked, and using the tamper detection calculating unit at the external

10 computer via the communicating unit with the external computer.

[Claim 11] A computer system, wherein

a tamper detecting program with a tamper detecting function according to

claim 1 is stored in a storage having a write protect function, and

the boot program comprises an executing unit of the tamper detecting

15 program.

[Claim 12] A computer system, wherein

a tamper detecting program with a tamper detecting function according to

claim 1 is stored in the auxiliary storage,

a value obtained by compressing the tamper detecting program by a one-way

20 function is stored in the ROM in advance,

the boot program comprises:

a tamper detecting program checking unit that checks presence or absence

of tamper of the tamper detecting program per se by compressing the tamper

detecting program by a one-way function and comparing a compressed value with the

25 compressed value stored; and

a tamper detecting program executing unit that executes the tamper detecting program.

[Claim 13] A computer system, wherein

5 a tamper detecting program with a tamper detecting function according to claim 1 is stored in the auxiliary storage,

the OS loader stored in a boot sector of the auxiliary storage is stored in advance in a portion other than the boot sector of the auxiliary storage, and

an alternate OS loader is stored in a boot sector of the auxiliary storage, the alternate OS loader comprising:

10 a tamper detecting program executing unit that executes reading of the tamper detecting program from the auxiliary storage; and

an OS loader executing unit that executes reading of the OS loader stored.

[0010]

15 [Problems to be Solved by the Invention] The problems of the conventional art are problems of operations.

[0011] This is because, as mentioned in the first example of the conventional art, for safer operations, a computer system is booted up by physically unrewritable safe mediums before check is executed. However, actually almost computer users
20 usually store OS or application programs in hard disks. A hard disk is not physically unrewritable medium and is likely to be tampered. Therefore, a computer system is booted up by a floppy disk which has OS and tamper checking program stored therein and which is write protected, and after checking OS and OS loader in the hard disk, the hard disk has to be booted up again. This bothers uses in operation and requires
25 unnecessary latency time of system boot-up.

[0012] An object of the present invention is to check with reliability presence or absence of tampered program to be executed at boot-up of the OS loader, Os or the computer under stress-free operation of users.

[0013] It is another object of the present invention to simplify the configuration
5 of a device for realizing the first object.

[0014] It is still another object of the present invention to increase the speed of tamper detecting processing.

[0015] It is still another object of the present invention to improve maintainability of the whole system.

10

[0033]

[Embodiments of the Invention]

[1] Descriptions of Configuration

Next, embodiments of the present invention are described in detail with
15 reference to the drawings.

[0034] Referring to Fig. 1, a first embodiment of the present invention comprises a controller 100 and an auxiliary storage 4.

[0035] The controller 100 comprises a ROM 1, a RAM 102 and a CPU 101 that executes programs stored in the ROM and the RAM.

20 [0036] A boot program 2 stored in the ROM 1 is a program of reading into the RAM 102 an OS loader 43 stored in the auxiliary storage 4 after initializing peripheral devices including the auxiliary storage 4 to start up a basic I/O system (hereinafter referred to as BIOS). The present invention provides this boot program 2 with a tamper detecting function.

25 [0037] The auxiliary storage 4 includes an OS file 44 and an OS loader 43 for

reading and executing the OS file 44 into the RAM 102. In the present invention, the auxiliary storage 4 has an arbitrary number of files which are to be set as files to be checked 42 of which presence or absence of a tamper is checked at boot-up of the system, and setting data, information for tamper check, for example digital signature information are stored in first stored information 41.

[0038] The above-mentioned tamper detecting function 3 comprises a stored information reading unit 31, a file to be checked reading unit 32 and a tamper detecting unit 33.

[0039] The stored information reading unit 31 reads the first stored information 41 from the auxiliary storage 4. The time of performing this processing, in other words, the start-up of the boot program 2 is to be performed before the boot-up of the OS, and therefore, it is impossible to access files via the OS. Accordingly, access to the auxiliary storage 4 has to be performed only with a function of low level BIOS. For example, when the OS is MS-DOS of Microsoft Corporation, directory data and a file allocation table (hereinafter referred to as FAT) of the auxiliary storage 4 are referred to so as to obtain information on physical location of each file stored in the auxiliary storage 4. In other words, a file name and a corresponding entry point of the FAT are written in the directory data, and a physical location of a stored file can be obtained by going around the FAT from the entry point. Hence, once a file name of the first stored information 41 is fixed in advance, the file name is used to obtain the physical location of the stored file in the auxiliary storage 4, and this may be performed by using the BIOS. The same hold true with other OS, in which a procedure of obtaining a physical location of a stored file executed by the OS is obtained to read the position from the BIOS.

[0040] The file to be checked reading unit 32 obtains contents of the file to be

checked 42 from the read first stored information 41 to read the file to be checked 42 from the auxiliary storage 4. Also in the file to be checked reading unit 32, access to the storage has to be taken only by using a function of low level of BIOS. Like in the above-mentioned stored information reading unit 31, a physical location in the auxiliary storage 4 is obtained in the same method as that a used OS executes to read the position from the BIOS.

[0041] The tamper detecting unit 33 checks presence or absence of tamper of a file to be checked read from the auxiliary storage 4 by using check information read from the first stored information 41 and second stored information 34 of a boot program per se. For example, a digital signature for each file to be checked is stored as first stored information 41 and a key for signature check is stored as second stored information 34. The tamper detecting unit 33 uses the digital signature and the key for signature check to check presence or absence of tamper of the file to be checked.

[0042] The tamper detecting function 3 can be also adopted in detecting presence or absence of tamper of the OS loader 43. The OS loader is stored in a portion called boot sector in the auxiliary storage 4. The boot sector is located at a physically head portion of the auxiliary storage, or if it is divided in plural partitions, at a head portion of each of the partitions. A drive name of a boot sector to be checked is written in a box storing a file name of the first stored information 41, thereby to specify a physical position in the auxiliary storage 4. Further, if data for checking tamper of the boot sector, for example, a digital signature is stored, the tamper detecting unit 33 can be used to check presence or absence of tamper of the boot sector or of the OS loader 43 stored therein. More specifically, since the OS loader 43 is a part of the boot sector, the presence or absence of tamper of the boot sector and the presence or absence of tamper of the OS loader 43 are different. However, since information on record

format and the like is stored in boot sectors other than that of the program of the OS loader 43, the boot sectors also may be subjected to tamper check. If the tamper check is limited to only the program portion, for example, a field for storing the number of first skipping bites and the number of end ignored bites has only to be added to the first stored information 41. Thus, only a program part to be checked is specified.

[0043] The tamper detecting function 3 can be also adopted in detecting presence or absence of tamper of the OS file 44. The OS file 44 is here a generic name for a basic OS program, a setting file which the OS program reads and various drivers to be executed at boot-up of the OS. Take, for example, MS-DOS of Microsoft Corporation, where the basic programs has file names of "io.sys", "msdos.sys" and "command.com", setting files which the OS programs reads include files of "config.sys" and "autoexec.bat" and the drivers, which are described in the setting files, are executed at the boot-up of the OS to be resident in a memory. If each of these file names is described in advance in a box for storing file names of the first stored information 41 and data for checking tamper, for example, a digital signature is described, the tamper detecting unit 33 is allowed to check presence or absence of tamper of files which are executed or referred to at boot-up of the OS.

[0044] The tamper detecting function 3 can be adopted in detecting presence or absence of tamper of the first stored information 41 per se. The purpose for checking presence or absence of tamper of the first stored information 41 per se is as follows. For example, if one file to be checked 42 is tampered, description added to the file to be checked 42 may be deleted from the first stored information 41 to avoid tamper from being checked so as to conceal the tamper. Check of the first stored information is useful for detecting presence or absence of concealment of the tamper. Specifically, as shown in Fig. 3 for example, tamper check stored information on

information obtained up to that time, for example a digital signature, is allocated to the end of the first stored information 41, and then, the tamper detecting unit 33 has only to utilize the tamper check stored information and the second stored information in the boot program 2 to check presence or absence of the tamper.

5 [2] Descriptions of Operation

Then, an operation of the embodiment of the present invention is described with reference to Figs. 1 and 2.

[0045] When a computer system is turned on, an execution address is set as a leading address of a boot program 2 stored in a ROM 1 so as to execute the boot
10 program 2.

[0046] Fig. 2 is a flowchart that shows a processing flow of the boot program 2. The boot program 2 first starts processing of initializing periphery devices (step A1). This step is for execution of test and initialization of connected periphery devices and initialization of a BIOS, thereby to make the BIOS usable. The steps A2 to A10 are
15 features of the present invention.

[0047] At first, at step A2, the above-mentioned stored information reading unit 31 is used to read the first stored information 41.

[0048] Then, at step A3, the first stored information 41 is subjected to tamper check. When this is performed using a digital signature, first, a portion other than the
20 own signature data of the first stored information 41 stored at the end of the first stored information 41 as shown in Fig. 3 is compressed by Hash function. Then, the own signature data of the first stored information 41 stored at the end thereof is decoded by a key stored in the second stored information 34 in the boot program 2. This compressed value and the decoded value are compared to check presence or
25 absence of tamper depending on coincidence of the comparison result with the

decoded value. If they are coincident, there has occurred no tampering and the processing proceeds to step A5. When they are not coincident, the processing proceed to step A8 (step A4).

[0049] Step A5 is a step of confirming that tamper check is performed for
5 each file to be checked 42 written in the first stored information 41. When all of the files to be checked 42 are not checked, the processing proceeds to step A6. When all the files to be checked are completed, the processing proceeds to step A11.

[0050] At step A6, the above-mentioned file to be checked reading unit 32 is used to read a file to be checked 42.

10 [0051] At step A7, the file to be checked 42 is subjected to tamper check. When tamper check is performed by using a digital signature, first, the file to be checked is compressed by Hash function. Then, a signature data stored in the first stored information 41 is decoded by a key which is stored in the second stored information 34 in the boot program 2. The resultant compressed value and the
15 decoded value are compared to check presence or absence of tamper depending on coincidence of the comparison result with the decoded value. If they are coincident, there has occurred no tampering and the processing proceeds to step A5. When they are not coincident, the processing proceeds to step A8 (step A4).

[0052] At step A4, when tampering is detected, presence of tampering is
20 displayed according to need in an additionally connected display, and it is determined whether the boot-up operation is continued or not (step A8, step A9). When continuation command is input at an additionally connected input device, it proceeds to step A5. When discontinue command is input, the boot program processing is terminated to abort the boot-up processing (step A10).

25 [0053] When at step A5, all the files to be checked are completely checked,

execution processing of the OS loader 43 is performed at step A11. The OS loader 43 written in a boot sector of the auxiliary storage 4 is read in the RAM 102 by the BIOS, an execution address of the CPU is set at the read RAM address, thereby to transfer the control to the OS loader 43. Then, the boot program processing is
5 finished.

[3] Other Embodiments of the Invention

Next, a second embodiment of the present invention will be described with reference to Fig. 4.

10 [Effects of the Invention] The first effect is in that presence or absence of tamper of an OS loader, an OS file or any other files can be detected with reliability and without users' consciousness before boot-up of the OS loader or OS. In other words, it is a boot program that performs the above-mentioned detecting, which is executed with reliability at boot-up of the computer system. Since the boot program is stored in the
15 ROM, there is no possibility of the program per se being tampered.

[0093] A second effect is in the simplicity of constituting a hardware configuration. In other words, the present invention can be configured by adding a soft module for performing tamper check to a boot ROM generally mounted in a system and by arranging files to be checked in physically sequential positions in an
20 auxiliary storage 4 and using a utility program for generating first stored information 41, and any special hardware does not have to be added.

[Brief Description of the Drawings]

[Fig. 1] A view that show a configuration of an embodiment of the present invention.

[Fig. 2] A flowchart that explains an operation of the present invention.

25 [Fig. 3] A view that explains an example of first stored information.

[Fig. 12] A view that explains the conventional art.

[Description of the Symbols]

	1	ROM
	2	Boot program
5	3	Tamper detecting function
	31	Stored information reading unit
	32	File to be checked reading unit
	33	Tamper detecting unit
	34	Second stored information
10	35	Stored information inputting unit
	36	Second stored information reading unit
	37	Communicating unit
	38	Tamper detecting program executing unit
	39	Tamper detecting program checking unit
15	391	Tamper detecting program compressed value
	4	Auxiliary storage
	41	First stored information
	42	File to be checked
	43	OS loader
20	44	OS file
	45	Boot sector
	46	Alternate OS loader
	461	Tamper detecting program executing unit
	462	OS loader executing unit
25	47	Tamper detecting program

	48	File to be checked list
	6	Input device
	*7	External storage
	71	Anti-tamper unit
5	8	External computer
	81	Anti-tamper unit
	82	Tamper detection calculating unit
	9	Tamper detecting program
	91	Write protect function storage
10	100	Controller
	101	CPU
	102	RAM

Fig. 1

15	100	Controller
	2	Boot program
	3	Tamper detecting function
	31	Stored information reading unit
	32	File to be checked reading unit
20	33	Tamper detecting unit
	34	Second stored information
	4	Auxiliary storage
	41	First stored information
	42	File to be checked
25	43	OS loader

44 OS file

Fig. 3

Leading marker

5 Whole size

Number of files to be checked (n)

First checked file name

10 Signature data of first checked file

Second checked file name

Signature data of second checked file

nth checked file name

Signature data of nth checked file

15 Own signature data of first stored information

Fig. 12

A

In preparation

20 Program file

Signed data

Signature generating process

Compressing by Hash function

Compressed value

25 Code by confidential key

Confidential key

B

In checking

Program file

5 Signed data

Signature verifying processing

Compressing by Hash function

Compressed value

Compare

10 Decoded value

Decode by public key

Public key

(19) 日本国特許庁 (J P) (12) 公開特許公報 (A) (11) 特許出願公開番号
特開平10-333902
 (43) 公開日 平成10年(1998)12月18日

(51) Int.Cl. ⁸	識別記号	F I
G 0 6 F 9/06	5 5 0	G 0 6 F 9/06 5 5 0 Z
9/445		12/14 3 1 0 Z
12/14	3 1 0	9/06 4 2 0 G

審査請求 有 請求項の数13 O L (全 14 頁)

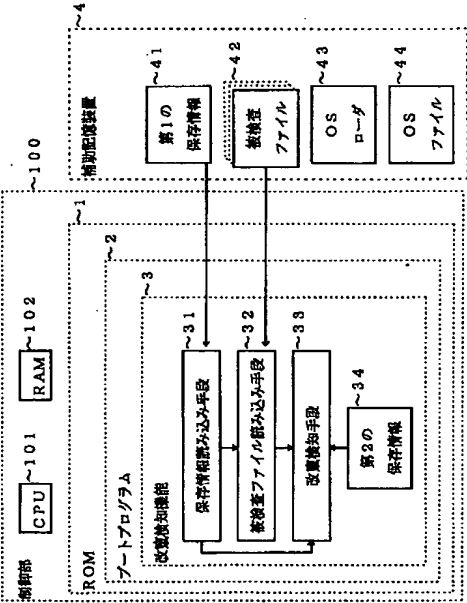
(21) 出願番号	特願平9-137181	(71) 出願人	394017491 株式会社エヌイーシー情報システムズ 神奈川県川崎市高津区坂戸3丁目2番1号
(22) 出願日	平成9年(1997)5月27日	(72) 発明者	望月 和広 神奈川県川崎市高津区坂戸3-2-1 株 式会社エヌイーシー情報システムズ内
		(74) 代理人	弁理士 京本 直樹 (外2名)

(54) 【発明の名称】 改ざん検知機能付きコンピュータシステム

(57) 【要約】

【課題】コンピュータの起動時に、OS、OSローダやその他の被検査ファイルの改ざんの有無を検査し、安全確認後にOS、OSローダ、アプリケーションを起動する。

【解決手段】ROM1に格納されたブートプログラム2は、BIOSの初期化の後に、補助記憶装置4に格納された第1の保存情報41をOSを介さずに読み込み、その読み込み内容にしたがって被検査ファイル42をOSを介さずに読み込む。これらの読み込み内容とROM1に格納された第2の保存情報とを用いて被検査ファイルの有無を検査する。改ざん無しを確認後に、OSローダ43を呼出してOS44を起動させる。



【特許請求の範囲】

【請求項1】起動するとまずROM内のブートプログラムが起動し、
前記ブートプログラムは補助記憶装置内のOSローダを起動し、
前記OSローダは前記補助記憶装置内のOSを起動するコンピュータシステムにおいて、
前記ブートプログラムが、
前記補助記憶装置に格納された第1の保存情報を読み込む保存情報読み込み手段と、
読み込んだ前記第1の保存情報の記載内容を元に前記補助記憶装置から任意個の被検査ファイルを読み込む被検査ファイル読み込み手段と、
前記ブートプログラムの内部に持つ第2の保存情報と読み込んだ前記第1の保存情報と前記各被検査ファイルとから前記各被検査ファイル個々の改ざんの有無を検査する改ざん検知手段と、を備えた改ざん検査機能を持つことを特徴としたコンピュータシステム。
【請求項2】前記第1の保存情報に前記各被検査ファイルに対するデジタル署名を格納し、
前記第2の保存情報に署名検証用の鍵を格納し、
請求項1に記載の改ざん検知手段が前記デジタル署名と前記署名検証用の鍵とで各被検査ファイルの改ざんの有無を検査することを特徴としたコンピュータシステム。
【請求項3】前記補助記憶装置に格納されたOSローダを、前記被検査ファイルの1つとして、ブート時に改ざんの有無の検査することを特徴としたコンピュータシステム。
【請求項4】前記補助記憶装置に格納されたOSファイルを、前記被検査ファイルの1つとして、ブート時に改ざんの有無の検査することを特徴としたコンピュータシステム。
【請求項5】前記第1の保存情報内に、自分自身の改ざん検査用の情報を持ち、
請求項1に記載の改ざん検知手段が、前記第1の保存情報の改ざんの有無を検査する機能をも有することを特徴としたコンピュータシステム。
【請求項6】前記第1の保存情報に前記各被検査ファイルを一方方向性関数で圧縮した圧縮値と、前記第1の保存情報に対するデジタル署名とを格納し、
前記第2の保存情報に署名検証用の鍵を格納し、
前記改ざん検知手段は、前記デジタル署名と前記署名検証用の鍵とで第1の保存情報の改ざんの有無を検査し、
前記圧縮値で各被検査ファイルの改ざんの有無を検査することを特徴としたコンピュータシステム。
【請求項7】前記ブートプログラムは、入力装置から前記第2の保存情報を取得する保存情報取得手段を備え、
前記第2の保存情報を使用者の入力から取得することを特徴とするコンピュータシステム。
【請求項8】請求項7において、

前記第1の保存情報に、前記各被検査ファイルと前記第2の保存情報とを一方方向性関数で圧縮した圧縮値と、
前記第1の保存情報自身と前記第2の保存情報とを一方方向性関数で圧縮した圧縮値と、を予め格納しておき、
前記改ざん検知手段は、請求項7に記載の前記保存情報取得手段で取得した前記第2の保存情報を用いて再度前記圧縮値を計算し、前記第1の保存情報に格納された前記圧縮値と比較することで、前記各被検査ファイルと前記第1の保存情報の改ざんの有無を検査することを特徴としたコンピュータシステム。
【請求項9】前記第2の保存情報を前記外部記憶装置に持ち、
前記ブートプログラムは前記外部記憶装置から第2の保存情報を取得する第2の保存情報読み込み手段を備え、
前記外部記憶装置は前記第2の保存情報の改ざん防止手段を備えることを特徴としたコンピュータシステム。
【請求項10】前記第2の保存情報を外部コンピュータに持ち、
前記外部コンピュータは、
前記第2の保存情報の改ざん防止手段と、
前記第2の保存情報を用いて改ざん検知に必要な計算を行なう改ざん検知計算手段とを備え、
前記ブートプログラムは、前記外部コンピュータとの通信手段を備え、
請求項1に記載の前記改ざん検知手段は、前記第1の保存情報と、前記各被検査ファイルと、前記外部コンピュータとの通信手段を介して前記外部コンピュータ側の改ざん検知計算手段を使用することとで、前記各被検査ファイル個々の改ざんの有無を検査することを特徴としたコンピュータシステム。
【請求項11】請求項1に記載の改ざん検知機能を持つ改ざん検知プログラムをライトプロテクト機能付きの記憶装置に格納し、
前記ブートプログラムは、前記改ざん検知プログラムの実行手段を備えることを特徴とするコンピュータシステム。
【請求項12】請求項1に記載の改ざん検知機能を持つ改ざん検知プログラムを前記補助記憶装置に格納し、
前記改ざん検知プログラムを一方方向性関数で圧縮した値を、予め前記ROM内に保存しておき、
前記ブートプログラムは、前記改ざん検知プログラムを一方方向性関数で圧縮し、その値を保存しておいた前記圧縮値と比較することにより、前記改ざん検知プログラム自身の改ざんの有無を検査する改ざん検知プログラム検査手段と、
前記改ざん検知プログラムを実行する改ざん検知プログラム実行手段と、を備えることを特徴とするコンピュータシステム。
【請求項13】請求項1に記載の改ざん検知機能を備えた改ざん検知プログラムを前記補助記憶装置に格納し、

さらに前記補助記憶装置のブートセクタに格納されている前記OSローダを予め前記補助記憶装置のブートセクタ以外の場所に保存しておき、前記改ざん検知プログラムを前記補助記憶装置から読み込み実行する改ざん検知プログラム実行手段と、保存された前記OSローダを読み込み実行するOSローダ実行手段と、を備えた代替OSローダを前記補助記憶装置のブートセクタに持たせることを特徴とするコンピュータシステム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、改ざん検知機能付きコンピュータシステムに関し、特にコンピュータシステムの起動時にファイルの改ざんの検知を行なう改ざん検知機能付きコンピュータシステムに関する。

【0002】

【従来の技術】従来、この種の改ざん検知機能付きコンピュータシステムは、例えば第1の例として「インテグリティチェック法を中心としたコンピュータウイルス対策システムの研究報告書」（1996年3月 情報処理振興事業協会発行）に示されるように、コンピュータウイルス（以下単にウイルスと呼ぶ）によるプログラムファイルの改ざんを検知するために用いられている。この報告書に示されているシステムはデジタル署名と呼ばれる暗号技術を用いてウイルスによるプログラムファイルの改ざんを検知している。この基本原理を図12を用いて説明する。

【0003】改ざんの有無を検査すべきプログラムファイルには予めデジタル署名（以下単に署名と呼ぶ）を生成しておく。これは署名生成プログラムで、まず検査すべきプログラムファイルをHash関数と呼ばれる関数で圧縮し、圧縮値を秘密鍵（以下秘密鍵と呼ぶ）で暗号化し、その暗号データを署名とする（図12A）。

【0004】検査時には、プログラムファイルを再度Hash関数で圧縮した値と、署名を公開された鍵（以下公開鍵と呼ぶ）で復号化した値とを比較する（図12B）。プログラムが改ざんされていなければこの2つの値は一致する。一致しなければ何らかの改ざんがあったと判断できる。

【0005】ここで用いる暗号方式は、非対称暗号あるいは公開鍵暗号と呼ばれるもので、暗号化する鍵と復号化する鍵が異なり、一方から他方が推測できない性質を持つものである。プログラムファイルの改ざんを隠蔽するために署名を偽造することは秘密鍵を知り得ない限り不可能である。

【0006】またHash関数とは入力データを160ビット程度に圧縮する一方方向性関数である。一方方向性関数とは、ある値に変換される元の値を逆算することが計算量的に不可能な関数を言う。従って改ざんを隠蔽するためにHash関数による圧縮値が変わらないように改ざ

ん内容を調整する必要があるが、これも不可能である。

【0007】前述の報告書付録5ページには、この従来技術の試作システム「IPAIncS」の運用上の注意が記述されている。そこには、OSと署名検証システムをライトプロテクトを施したFDで保持し、そのFDでコンピュータを立ち上げてから実行する方法が安全だとしている。これはOSローダやOSファイル、もしくは署名検証システム自身がウイルスに感染した場合に、署名検証システムで改ざんの有無を検査をする以前にウイルスが動作してしまうからである。このウイルスがステルス機能を持つものであれば、後から感染（つまり改ざん）されたプログラムファイルを検査しても、改ざんは隠蔽されて検知できない。ステルス機能が検査のためにプログラムファイルを読み込むのを事前に察知し、改ざん部分を修復した読み込み結果を渡すからである。

【0008】我が国におけるコンピュータウイルス感染被害の報告の半数以上が、OSローダに感染するブートセクタ感染ウイルスである現在、前記のような運用方法をとる必要性は高い。

【0009】プログラムファイルの改ざんを防止するシステムの例としては、特開平6-230959号公報に示されるものがある。このシステムでは、専用の拡張ボードを用いて、機能を拡張したBIOSとDOSを動作させることにより、プログラムファイルが不正に書き換えられることを防止している。しかしこのシステムは、万が一プログラムファイルの改ざんがあった後にそれを検知できる仕組みではなく、本発明とは目的が異なるので、本方式の代替の方法となるものではない。しかも、ウイルスが感染のためにプログラムファイルを改ざんするのを監視するものであり、任意のファイル（例えばセキュリティの設定や暗号鍵などを格納したデータファイル）の改ざんを防止するものではない。さらに、不正にはアクセスできないROMとRAMを持つハードウェア構成を必須とする装置構成上の制約がある。

【0010】

【発明が解決しようとする課題】従来技術の問題点は、操作性の問題である。

【0011】その理由は、前記従来の技術の第1の例で示したように、安全に運用するためには、物理的に書き換え不能な安全な媒体でコンピュータシステムを立ち上げて検査を実行しなければならない。しかし実際問題としては、ほとんどのコンピュータ使用者は通常、OSやアプリケーションプログラムはハードディスクに保持している。ハードディスクは物理的に書き換え不能な媒体ではなく、改ざんの危険性が高い。従ってOSと改ざん検査プログラムの格納されたライトプロテクトを施したフロッピーディスクで立ち上げ、ハードディスク内のOSやOSローダを検査した後に、再度ハードディスクで立ち上げると言った方法を取らざるを得ない。これは使用者に操作の煩わしさと、余分なシステムの立ち上げ待

10

20

30

40

50

ち時間を強いることになる。

【0012】本発明の主たる目的は、使用者にストレスを感じさせない操作性のもとで、OSローダやOSや他のコンピュータ起動時に実行されるプログラムの改ざんの有無を確実に検査することである。

【0013】本発明の他の目的は、第1の目的を実現するための装置構成を簡素化する点である。

【0014】本発明の他の目的は、改ざん検知処理の高速化を図る点である。

【0015】本発明の他の目的は、システム全体の保守性を向上させる点である。

【0016】

【課題を解決するための手段】本発明の第1の実施形態では、ROM1に格納されたブートプログラム2に、補助記憶装置4に格納された任意個の任意の被検査ファイルの改ざんの有無の検査する機能を持たせる。より具体的には、補助記憶装置4に格納された第1の保存情報41を読み込む保存情報読み込み手段(図1の31)と、読み込んだ第1の保存情報41の記載内容を元に補助記憶装置4から任意個の被検査ファイル42を読み込む被検査ファイル読み込み手段(図1の32)と、ブートプログラム2の内部に持つ第2の保存情報34と読み込んだ第1の保存情報41と各被検査ファイル42とから各被検査ファイル個々の改ざんの有無を検査する改ざん検知手段(図1の33)とを有する。

【0017】本発明の第2の実施形態では、第1の実施形態で記載した第2の保存情報34を使用者の入力から取得することを特徴とする。より具体的には、ブートプログラム2が入力装置6から第2の保存情報34を取得する保存情報取得手段(図5の35)とを有する。

【0018】本発明の第3の実施形態では、第2の実施形態において一方向性関数による圧縮値で改ざんの有無を検査する特徴を持つ。より具体的には、入力装置6から第2の保存情報34を取得する保存情報取得手段(図5の35)と、被検査ファイル42と第2の保存情報34とを一方向性関数で圧縮した圧縮値と予め第1の保存情報41に保存された圧縮値とを比較して改ざんの有無を判定する改ざん検知手段(図5の33)とを有する。

【0019】本発明の第4の実施形態では、第2の保存情報34を外部記憶装置7に持つことを特徴とする。より具体的には、ブートプログラム2は外部記憶装置7から第2の保存情報34を取得する第2の保存情報読み込み手段(図7の36)とを有し、外部記憶装置7は第2の保存情報34の改ざん防止手段(図7の71)とを有する。

【0020】本発明の第5の実施形態では、第2の保存情報34を外部コンピュータ8に持ち、改ざん検知に必要な計算処理を外部コンピュータ8側で行なわせる。より具体的には、外部コンピュータ8は、第2の保存情報34の改ざん防止手段(図8の81)と、改ざん検知に

必要な計算を行なう改ざん検知計算手段(図8の82)とを備え、ブートプログラム2は、外部コンピュータ8との通信手段(図8の37)と、通信手段37を介して外部コンピュータ8側の改ざん検知計算手段82を使用して各被検査ファイル個々の改ざんの有無を検査する改ざん検知手段(図8の33)とを有する。

【0021】本発明の第6の実施形態では、改ざん検知機能3を改ざん検知プログラム9をライトプロテクト機能付きの記憶装置91に保持してブート時に実行する。より具体的には、ブートプログラム2は、改ざん検知プログラム9の実行手段(図9の38)とを有する。

【0022】本発明の第7の実施形態では、改ざん検知機能3を改ざん検知プログラム9を補助記憶装置4に保持してブート時に検査して実行する。より具体的には、ブートプログラム2は、一方向性関数を用いて改ざん検知プログラム9の改ざんの有無を検査する改ざん検知プログラム検査手段(図10の39)と、改ざん検知プログラム9を実行する改ざん検知プログラム実行手段(図10の38)とを有する。

【0023】本発明の第8の実施形態では、OSローダ43に改ざん検査機能3を持たせる。より具体的には、代替OSローダ46が、改ざん検知機能3を備えた改ざん検知プログラム9を前記補助記憶装置4から読み込み実行する改ざん検知プログラム実行手段(図11の461)と、別途保存されたOSローダ43を読み込み実行するOSローダ実行手段(図11の462)とを有する。

【0024】

【作用】本発明の第1の実施形態では、保存情報読み込み手段は、被検査ファイル読み込み手段に対して、被検査ファイルを読み込むための情報、具体的には補助記憶装置内に被検査ファイルが格納されている物理的位置の情報を提供する。また、改ざん検知手段に対しては、被検査ファイルの改ざんの有無を判定するための情報、具体的には被検査ファイルに対する署名データなどを提供する。

【0025】被検査ファイル読み込み手段は、改ざん検知手段に対して、現在の被検査ファイルの内容を提供する。

【0026】改ざん検知手段は、提供された被検査ファイルの改ざんの有無を判定するための情報と、現在の被検査ファイルの内容と、ブートプログラム2の内部に持つ第2の保存情報(具体的には署名検証用の鍵など)から、被検査ファイルの改ざんの有無を検査する。

【0027】本発明の第2の実施形態と第3の実施形態では、保存情報取得手段が、改ざん検知手段に、第2の保存情報を提供する。

【0028】本発明の第4の実施形態では、外部記憶装置に保存された第2の保存情報は、外部記憶装置の改ざん防止手段で改ざんから守られる。保存情報読み込み手

段は第2の保存情報を外部記憶装置から取得し、改ざん検知手段に提供する。

【0029】本発明の第5の実施形態では、外部コンピュータに保存された第2の保存情報は、外部外部コンピュータの改ざん防止手段で改ざんから守られる。ブートプログラムは周辺機器の初期化処理後、改ざん検知手段で、通信手段を介して外部コンピュータとデータの送受信を行ない、外部コンピュータ側の改ざん検知計算手段を利用して、各被検査ファイルの改ざんの有無を検査する。

【0030】本発明の第6の実施形態では、ブートプログラムは周辺機器の初期化処理後、改ざん検知プログラム実行手段で、ライトプロテクト機能付きの記憶装置に格納された改ざん検知プログラムを実行する。

【0031】本発明の第7の実施形態では、ブートプログラムは周辺機器の初期化処理後、改ざん検知プログラム検査手段で、補助記憶装置4に格納された改ざん検知プログラムに改ざんがないかチェックする。次いで、改ざん検知プログラム実行手段で改ざん検知プログラムを実行する。これで被検査ファイルを検査する。

【0032】本発明の第8の実施形態では、ブートプログラムにより代替OSローダが実行され、代替OSローダはまず、改ざん検知プログラム実行手段で改ざん検査機能を持った改ざん検知プログラムを実行する。これでこれから起動するOSに必要な被検査ファイルを検証する。次いで、代替OSローダ46は、OSローダ実行手段で、別途保存されたOSローダ43を読み込み実行し、OSが起動する。

【0033】

【発明の実施の形態】

【1】構成の説明

次に、本発明の実施の形態について図面を参照して詳細に説明する。

【0034】図1を参照すると、本発明の第1の実施の形態は、制御部100と、補助記憶装置4を含む。

【0035】制御部100は、ROM1とRAM102と、それらに格納されたプログラムを実行するCPU101を含む。

【0036】ROM1に格納されたブートプログラム2は、補助記憶装置4を含む周辺機器の初期設定を行ない基本入出力システム（以下BIOSと略す）を使用可能にした後、補助記憶装置4に格納されたOSローダ43をRAM102に読み込み実行するプログラムである。本発明は、このブートプログラム2に、改ざん検知機能3を持たせることを特徴とする。

【0037】補助記憶装置4には、OSファイル44と、OSファイル44をRAM102に読み込み実行するためのOSローダ43とが含まれている。本発明では、この補助記憶装置4に任意個の任意のファイルを、システム立ち上げ時に改ざんの有無を検査する被検査フ

ァイル42として設定でき、その設定内容と、改ざん検査用の情報、例えばデジタル署名などの情報を第1の保存情報41に格納していることを特徴としている。

【0038】前述の改ざん検知機能3は、保存情報読み込み手段31と、被検査ファイル読み込み手段32と、改ざん検知手段33とを備える。

【0039】保存情報読み込み手段31は、補助記憶装置4から第1の保存情報41を読み込むものである。この処理を行なう時点、つまりブートプログラム2が起動する時点は、まだOSの起動前であり、OSを介したファイルアクセスは使用できない。従ってBIOSの低レベルな関数だけを使って補助記憶装置4をアクセスする必要がある。例えば、OSがマイクロソフト社製のMS-DOSであれば、補助記憶装置4のディレトリデータとファイルアロケーションテーブル（以下FATと略す）を参照して、各ファイルの格納された補助記憶装置4内での物理的位置の情報を得ることができる。つまりディレトリデータにはファイル名とFATのエントリポイントの対応が記述されており、そのエントリポイントから順にFATを辿ることにより、ファイルが格納された物理的格納位置を知ることができる。従って予め第1の保存情報41のファイル名を固定しておけば、そのファイル名から補助記憶装置4内の物理的格納位置を取得することができ、BIOSを使って読み出せばよい。他のOSの場合も同様に、そのOSが行なうファイルの物理的格納位置の取得方法を実行し、BIOSで読み出せばよい。

【0040】被検査ファイル読み込み手段32は、読み込んだ第1の保存情報41から被検査ファイル42の設定内容を取得し、その被検査ファイル42を補助記憶装置4から読み込むものである。ここでもBIOSの低レベルな関数だけを使っての記憶装置にアクセスする必要がある。これも前述の保存情報読み込み手段31と同様に、使用するOSが行なうのと同じ方法で補助記憶装置4内の物理的格納位置を取得し、BIOSを使って読み出せばよい。

【0041】改ざん検知手段33は、補助記憶装置4から読み込んだ各被検査ファイルに対して、第1の保存情報41から読み取った検査用の情報と、ブートプログラム自身が持つ第2の保存情報34とから、改ざんの有無を検査する。例えば、第1の保存情報41として、各被検査ファイルに対するデジタル署名を格納し、第2の保存情報34として、署名検証用の鍵を格納しておく。改ざん検知手段33は前記デジタル署名と前記署名検証用の鍵とで各被検査ファイルの改ざんの有無を検査する。

【0042】また、この改ざん検知機能3で、OSローダ43の改ざんの有無を検証することもできる。OSローダは補助記憶装置4のブートセクタと呼ばれる部分に格納されている。ブートセクタは補助記憶装置の物理的

頭、もしくは複数のパーティションが切られているときは各パーティションの先頭に位置する。第1の保存情報41のファイル名の格納欄に検査したいブートセクタのドライブ名を記述しておけば、その補助記憶装置4内での物理的位置は特定できる。更にブートセクタに対する改ざん検査用のデータ、例えばデジタル署名を格納しておけば、この改ざん検知手段33で、ブートセクタの改ざんの有無、つまりはその中に格納されたOSロード43の改ざんの有無を検査できる。厳密にはOSロード43はブートセクタの一部であるので、ブートセクタの改ざんの有無と、OSロード43の改ざんの有無は異なるが、OSロード43のプログラム部分以外のブートセクタには記録フォーマットなどの情報が格納されているので、ここも含めた改ざんの検査をすれば問題ない。プログラム部分だけに限定させるなら、例えば、第1の保存情報41に先頭の読み飛ばしバイト数と終端の無視するバイト数を格納するフィールドを追加すれば、検査すべきプログラム部分のみを特定できる。

【0043】改ざん検知機能3でOSファイル44の改ざんの有無を検証することもできる。OSファイル44とは、ここでは基本となるOSプログラムと、それが読み込む設定ファイルと、およびOS起動時に実行される各種ドライバ類の総称とする。例えばマイクロソフト社のMS-DOSを例にとると、基本となるOSプログラムはファイル名が「io.sys」、「msdos.sys」、「command.com」などのファイルで、それらが読み込む設定ファイルは「config.sys」や「autoexec.bat」などのファイルで、ドライバ類は前記設定ファイルに記述されたものがOS起動時に実行されメモリ中に常駐する。これらのファイル名を予め第1の保存情報41のファイル名の格納欄に記述し、さらに改ざん検査用のデータ、例えばデジタル署名を記述しておけば、前記改ざん検知手段33でOSの起動時に実行もしくは参照されるファイルの改ざんの有無を検査できる。

【0044】改ざん検知機能3で第1の保存情報41自身の改ざんの有無を検証することもできる。この第1の保存情報41自身の改ざんの有無を検査する目的は、次の通りである。例えば、ある被検査ファイル42が改ざんされた際に、その被検査ファイル42に対する記述を第1の保存情報41から削除してしまい、検査を逃れて改ざんを隠蔽されることが有り得る。この第1の保存情報の検証は、この隠べいの事実の有無を検出するのに有益である。具体的な方法としては、例えば図3で示すように第1の保存情報41の末尾に、そこまでの情報に関する改ざん検査用保存情報、例えばデジタル署名を付与しておき、改ざん検知手段33は、この改ざん検査用保存情報とブートプログラム2内の第2の保存情報とを用いて改ざんの有無を検査すればよい。

【2】 動作の説明

次に、図1および図2を参照して、本発明の実施の形態の動作について説明する。

【0045】コンピュータシステムの電源を入れると、CPU101の実行アドレスはROM1に格納されたブートプログラム2の処理の先頭アドレスにセットされ、ブートプログラム2の実行が開始される。

【0046】図2はブートプログラム2の処理の流れを示すフローチャートである。ブートプログラム2はまず周辺機器の初期化の処理を実行する（ステップA1）。これは、接続された周辺機器のテストや初期設定の実行と、BIOSの初期設定などを行なうものである。これによりBIOSが使用可能になる。ステップA2からステップA10までが本発明の特徴となる部分である。

【0047】まずステップA2で前述の保存情報読み込み手段31を用いて、第1の保存情報41を読み込む。

【0048】次いでステップA3で第1の保存情報41自身の改ざんの検査を行なう。これは例えばデジタル署名を使用する方法であれば、まず図3で示すように第1の保存情報41の末尾に格納された第1の保存情報41自身の署名データを除いた部分をHash関数で圧縮する。次いで末尾にある第1の保存情報41自身の署名データを、ブートプログラム2内の第2の保存情報34に格納された鍵を使って復号する。この圧縮値と復号値と比較して、一致するか否かで改ざんの有無を検査する。一致していれば改ざんは無いので、ステップA5に進み、一致していなければステップA8に進む（ステップA4）。

【0049】ステップA5は、第1の保存情報41に書かれた被検査ファイル42の数だけ改ざんの有無を検査したどうかを確認するためのステップである。全部の被検査ファイル42の検査が終わっていない場合、ステップA6へと進み、全部の検査が終わっていた場合はステップA11へと進む。

【0050】ステップA6は前述の被検査ファイル読み込み手段32を用いて、各被検査ファイル42を読み込む。

【0051】次いでステップA7で、被検査ファイル42の改ざんの検査を行なう。これは例えばデジタル署名を使用する方法であれば、まず被検査ファイルをHash関数で圧縮する。次いで第1の保存情報41に格納された署名データを、ブートプログラム2内の第2の保存情報34に格納された鍵を使って復号する。この圧縮値と復号値と比較して、一致するか否かで改ざんの有無を検査する。一致していれば改ざんは無いので、ステップA5に進み、一致していなければステップA8に進む（ステップA4）。

【0052】ステップA4において、改ざんが検知された場合には、必要に応じて別途接続された表示装置に表示し、継続して起動処理を継続するか確認する（ステップA8、ステップA9）。別途接続された入力装置から

継続を指示された場合はステップA5へと進み、中止が指示された場合はブートプログラムの処理を終了させ、起動処理を中止する(ステップA10)。

【0053】ステップA5で全ての被検査ファイルを検査を終了した場合、ステップA11でOSローダ43の実行処理を行なう。これは補助記憶装置4のブートセクタに書かれたOSローダ43をBIOSを使ってRAM102に読み込み、読み込まれたRAMアドレスにCPUの実行アドレスをセットすることにより、OSローダ43に制御を移す。これでブートプログラムの処理は終了する。

【3】発明の他の実施形態

次に本発明の第2の実施形態について図4を参照して説明する。

【0054】図4は、第2の実施形態における、第1の保存情報41の構成を示したものである。

【0055】この実施形態では、第1の保存情報41に各被検査ファイル42を一方方向性関数で圧縮した圧縮値と、第1の保存情報41自身に対するデジタル署名とを格納しておく。このデジタル署名の検証用の鍵は、第1の実施形態と同じくブートプログラム2内の第2の保存情報34に格納しておく。

【0056】この実施形態におけるブートプログラム2の改ざん検知手段33は、第1の実施形態と同じく、まず第1の保存情報41自身に対する署名データで、第1の保存情報41自身に改ざんが無いことを確認する。次いで被検査ファイル読み込み手段32で読み込んだ被検査プログラムを一方方向性関数で圧縮し、その圧縮値と第1の保存情報41に格納された圧縮値とを比較する。被検査ファイル42に改ざんが無い場合はこの値は一致する。被検査ファイル42に改ざんがあった場合、現在の内容を圧縮した値と保存された圧縮した値は異なるので、改ざんを検知できる。

【0057】上記の圧縮値は使用している一方方向性関数の仕様さえ判れば生成することができる。従って被検査プログラム42が改ざんされ、その改ざんを隠蔽するため、第1の保存情報41内に保存された圧縮値までも、改ざん結果の圧縮値に書き換えられてしまうこともありうる。この場合、前述の圧縮値同士の比較は一致し、検査をパスされてしまうが、その前に第1の保存情報41自身の改ざん検査で、改ざんの隠蔽があったことを検知できる。

【0058】この実施形態においては、前述のように第1の保存情報41自身までもが改ざんされ、被検査ファイル42に改ざんが隠蔽された場合に、被検査ファイル42のいずれかが改ざんされそれが隠蔽されたと検知できるものの、被検査ファイル42のどれが改ざんされたのかの特定がこの方法だけではできないというデメリットはある。ただし署名復号の処理が1回で済むため全ての被検査ファイル42を検証する処理時間は短時間で済

む。従って処理速度を重視する場合には有用である。

【0059】次に本発明の第3の実施形態について図5を参照して説明する。

【0060】図5は、ブートプログラム2の改ざん検知機能3の一部として、入力装置6を用いて前述の第2の保存情報34を使用者の入力操作により取得し、改ざん検知手段33で用いることを特長としている。

【0061】ブートプログラム2は、図2のステップA1の周辺機器の初期化の後で、入力装置6から第2の保存情報34を取得する。これはステップA1で入力装置6の初期化とBIOSの初期化が完了しているので、BIOSを使って入力装置6から入力情報を受け取ればよい。

【0062】この実施形態は、第2の保存情報34を使用者の入力操作により取得するため、大きなサイズの情報を使用できないというデメリットがあるものの、第2の保存情報34をブートプログラム2の格納されたROM1から切り離すことで、情報を更新できるという効果がある。さらに第2の保存情報34をROM1から切り離すことにより、改ざんを企てる第三者から秘匿することができる。その結果次の第4の実施形態が可能になる。

【0063】次に本発明の第4の実施形態について図6を参照して説明する。

【0064】図6は、前記第4の実施形態における、第1の保存情報42の例である。

【0065】この例では、第1の保存情報41に、各被検査ファイル42と使用者だけが知り得る第2の保存情報とを連結し、一方方向性関数で圧縮した圧縮値を格納する。ここでいう連結とは、各被検査ファイル42の末尾に第2の保存情報を繋げることを言う。連結する以外にも、2つの情報の排他的論理和(XOR)を取る方法でも構わない。第1の保存情報41自身も、第2の保存情報34と連結して一方方向性関数で圧縮した圧縮値を、自分自身の末尾に持たせておく。

【0066】改ざん検知手段は、検査時に、前記保存情報取得手段で取得した前記第2の保存情報とを連結し、圧縮値を計算し、第1の保存情報41に格納された値と比較することで、前記各被検査ファイルの改ざんの有無を確認する。第1の保存情報41自身の改ざんの有無も同様に確認する。

【0067】被検査プログラムの改ざんを隠蔽するために、この保存情報を偽造することは、使用者だけが知り得る第2の保存情報を知り得ない限り、一方方向性関数の性質上、計算量的に困難である。

【0068】この実施形態は、前記第3の実施形態の効果に加え、暗号処理を含まないため処理の高速化と、ブートプログラム2が簡素化できる効果がある。

【0069】次に本発明の第5の実施形態について図7を参照して説明する。

【0070】この実施形態では、第2の保存情報34を外部記憶装置7に持ち、ブートプログラム2は外部記憶装置7から第2の保存情報を取得する第2の保存情報読み込み手段36を備え、なおかつ外部記憶装置7は第2の保存情報34の改ざん防止手段71を備えることを特徴としている。

【0071】外部記憶装置7は例えば、フロッピーディスクとそのドライブで構成することができる。この場合は、第2の保存情報読み込み手段36は、BIOSを用いてドライブをアクセスすることで第2の保存情報34を取得できる。改ざん防止手段71は、フロッピーディスクにライトプロテクトを施せばよい。

【0072】この実施形態は、第2の保存情報34をブートプログラム2の格納されたROM1から切り離すことで、保存情報の更新が容易にできるという効果がある。

【0073】次に本発明の第6の実施形態について図8を参照して説明する。

【0074】この実施例では、第2の保存情報34を外部コンピュータ8に持ち、この外部コンピュータは、第2の保存情報の改ざん防止手段81と、第2の保存情報を用いて改ざん検知に必要な計算を行なう改ざん検知計算手段82とを備える。さらにブートプログラム2は、外部コンピュータ8との通信手段37を備える。そして、ブートプログラム2の改ざん検知手段33は、第1の保存情報41と各被検査ファイル42と、外部コンピュータとの通信手段37を介して外部コンピュータ8側の改ざん検知計算手段82を使用することと、各被検査ファイル42個々の改ざんの有無を検査するものとする。

【0075】この実施形態でいう外部コンピュータ8は、例えば、公開鍵暗号の計算機能を備えたICカードとそのリーダ／ライタとで構成できる。

【0076】公開鍵暗号の計算機能を備えたICカードとしては、CP8トランザック(Transac)社製のTB98Sなどが知られている。このようなカードを用いれば、前述の改ざん検知計算手段82として、署名の復号化処理をブートプログラム2から切り離し、ICカード側で行なうことができる。この場合ブートプログラム2側の改ざん検知手段33は、被検査ファイル42の署名をICカードに送り、その復号値を受け取り、その値とブートプログラム2側で圧縮した値とを比較して改ざんを検査すればよい。

【0077】外部コンピュータ2にICカードを用いた場合の第2の保存情報の改ざん防止手段81は、所定の手続き、例えば使用者のPINコードを照合しない限りデータの更新を許可しない機能で実現できる。この機能はICカードが一般的に持つ機能である。

【0078】また、外部コンピュータ2にICカードを用いた場合のブートプログラム2側の通信手段37は、

例えばRS232-Cポートを介して接続されるリーダ／ライタを使えば、実現できる。これは、ブートプログラム2は最初に周辺機器を初期化する(図2ステップA1)ので、検査を開始する時点ではRS232-C関連のBIOSも使用可能であり、これを用いて通信を行えばよい。

【0079】この実施形態においては、改ざん検知に関する計算処理の一部をROM1内のブートプログラムから切り離すことで、計算処理の変更やバージョンアップなどに対応しやすくなる効果がある。

【0080】次に本発明の第7の実施形態について図9を参照して説明する。

【0081】この実施形態では、改ざん検知機能3を持つ改ざん検知プログラム9をライトプロテクト機能付きの記憶装置91に格納する。

【0082】ライトプロテクト機能付きの記憶装置91は、拡張ボードとそれに搭載された拡張ROMとで構成することができる。多くの拡張ボードには、そのボードを扱うためのBIOSが搭載された拡張ROMをボード上に持つ。これに対応すべく、多くのコンピュータシステムのブートプログラムは、BIOSが格納された拡張ROMが装着されていないか拡張ROM用のアドレス空間を検索し、見つけた場合はその初期ルーチンと呼び出す機能を持つ。改ざん検知プログラム9を拡張ボード上の拡張ROMに格納すれば、前述の初期ルーチンと呼び出す機能がそのまま、改ざん検知プログラム実行手段38となる。

【0083】ライトプロテクト機能付きの記憶装置91は、他にフロッピーディスクとそのドライブ装置とでも構成できる。つまり、ブートプログラム2が記憶装置91から改ざん検知プログラム9を読み込む方法は、例えば改ざん検知プログラム9を記憶装置91の所定の物理的位置から始まる物理的連続領域に格納しておき、その所定の位置からBIOSを使って読み出せばよい。実行する方法は、ジャンプ命令でCPU100のIPアドレスを、RAM102に読み込まれた改ざん検知プログラム9の先頭アドレスに設定すればよい。

【0084】この実施形態においては、改ざん検知機能3をROM1内のブートプログラムから切り離すことで、検知方法の変更やバージョンアップなどに対応しやすくなる効果がある。

【0085】次に本発明の第8の実施形態について図10を参照して説明する。

【0086】この実施形態では、改ざん検知機能3を持つ改ざん検知プログラム9を補助記憶装置4に格納し、ROM1内のブートプログラム2に、改ざん検知プログラム9を予め一方方向性関数で圧縮した改ざん検知プログラム圧縮値391を記録しておく。そしてブート時には、現在の被検査プログラムの内容を一方方向性関数で圧縮して値と、保存された改ざん検知プログラム圧縮値391

とを比較して、改ざん検知プログラム9自身の改ざんの有無を検査する(改ざん検知プログラム検査手段39)。改ざん検知プログラム9を読み込む方法は、例えば改ざん検知プログラム9を記憶装置91の所定の物理的位置から始まる物理的連続領域に格納しておき、その所定の位置からBIOSを使ってRAM102に読み出せばよい。改ざん検知プログラム実行手段38は、ジャンプ命令でCPU100のIPアドレスを、RAM102に読み込まれた改ざん検知プログラム9の先頭アドレスに設定すればよい。

【0087】この実施形態では、本発明を実施するためにROM1に格納されたブートプログラム2が新たに備える機能が、方向性関数を用いた簡易な改ざん検知プログラム検査手段39と、改ざん検知プログラム9をロードして実行する改ざん検知プログラム実行手段38だけであるので、ブートプログラム2のプログラムサイズを小さくでき、占有するROM1のメモリ空間を小さくできる効果がある。

【0088】次に本発明の第9の実施形態について図11を参照して説明する。

【0089】この実施形態では、改ざん検知機能3を備えた改ざん検知プログラム9を補助記憶装置4から読み込み実行する改ざん検知プログラム実行手段461と、補助記憶装置4内のブートセクタ以外の物理的位置に別途保存されたOSローダ43を読み込み実行するOSローダ実行手段462とを備えた代替OSローダ46を、補助記憶装置4のブートセクタに持たせる。

【0090】改ざん検知プログラム実行手段461は、例えば、改ざん検知プログラム9を補助記憶装置4の所定の物理的位置から始まる物理的連続領域に格納しておき、その所定の位置からBIOSを使ってRAM102に読み込み、ジャンプ命令でCPU100のIPアドレスを、RAM102に読み込まれた改ざん検知プログラム9の先頭アドレスに設定すればよい。

【0091】OSローダ実行手段462も同様に構成できる。つまり、別途保存したOSローダも補助記憶装置4の所定の物理的位置から始まる物理的連続領域に格納しておき、改ざん検知プログラム実行手段461と同様に読み込み、実行すればよい。

【0092】この実施形態では、補助記憶装置4に複数のOSをインストールして使い分け使用環境において、起動するOS、つまりは起動するブートセクタが指定された後に改ざん検知機能3が動作することを特徴とする。これにより起動するOSに必要なファイルだけ、改ざんの検査をさせることができ、検査処理時間の短縮が図れる。

【発明の効果】第1の効果は、OSローダやOSが起動する前に、OSローダやOSファイル、あるいはその他任意のファイルの改ざんの有無を、確実に、しかも使用者は意識せずに、検証することができることである。そ

の理由は、上記の検証を行なうのはブートプログラムであり、当該コンピュータシステム起動時には確実に実行されるものであり、しかもROMに格納されているのでこれ自身が改ざんされる心配はないためである。

【0093】第2の効果は、ハードウェア構成が簡素に構成できることである。その理由は、通常システムが持つブートROMに改ざん検査を行なうソフトモジュールを追加することと、被検査ファイルを補助記憶装置4の物理的連続領域に配置し第1の保存情報41を生成するユーティリティプログラムを使用するだけで、本発明を構成することができ、特別なハードウェアの追加を必須としないためである。

【0094】

【図面の簡単な説明】

【図1】本発明の一実施形態の構成を示す図である。

【図2】本発明の動作を説明するためのフローチャートである。

【図3】第1の保存情報の一例を説明するための図である。

【図4】第1の保存情報の他の例を説明するための図である。

【図5】本発明の他の実施の形態の構成を表わす図である。

【図6】第1の保存情報の他の例を説明する図である。

【図7】本発明の他の実施の形態の構成を表わす図である。

【図8】本発明の他の実施の形態の構成を表わす図である。

【図9】本発明の他の実施の形態の構成を表わす図である。

【図10】本発明の他の実施の形態の構成を表わす図である。

【図11】本発明の他の実施の形態の構成を表わす図である。

【図12】従来の技術を説明するための図である。

【符号の説明】

1 ROM

2 ブートプログラム

3 改ざん検知機能

40 31 保存情報読み込み手段

32 被検査ファイル読み込み手段

33 改ざん検知手段

34 第2の保存情報

35 保存情報入力手段

36 第2の保存情報読み込み手段

37 通信手段

38 改ざん検知プログラム実行手段

39 改ざん検知プログラム検査手段

391 改ざん検知プログラム圧縮値

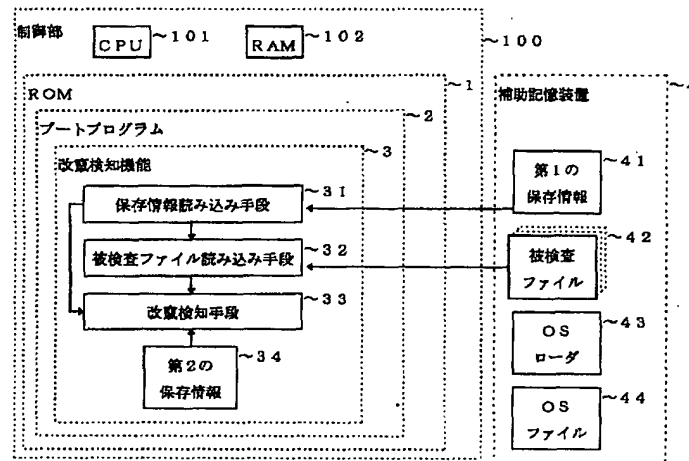
50 4 補助記憶装置

41 第1の保存情報
 42 被検査ファイル
 43 OSローダ
 44 OSファイル
 45 ブートセクタ
 46 代替OSローダ
 461 改ざん検知プログラム実行手段
 462 OSローダ実行手段
 47 改ざん検知プログラム
 48 被検査ファイルリスト
 6 入力装置

* 7 外部記憶装置
 71 改ざん防止手段
 8 外部コンピュータ
 81 改ざん防止手段
 82 改ざん検知計算手段
 9 改ざん検知プログラム
 91 ライトプロテクト機能付き記憶装置
 100 制御部
 101 CPU
 102 RAM

*

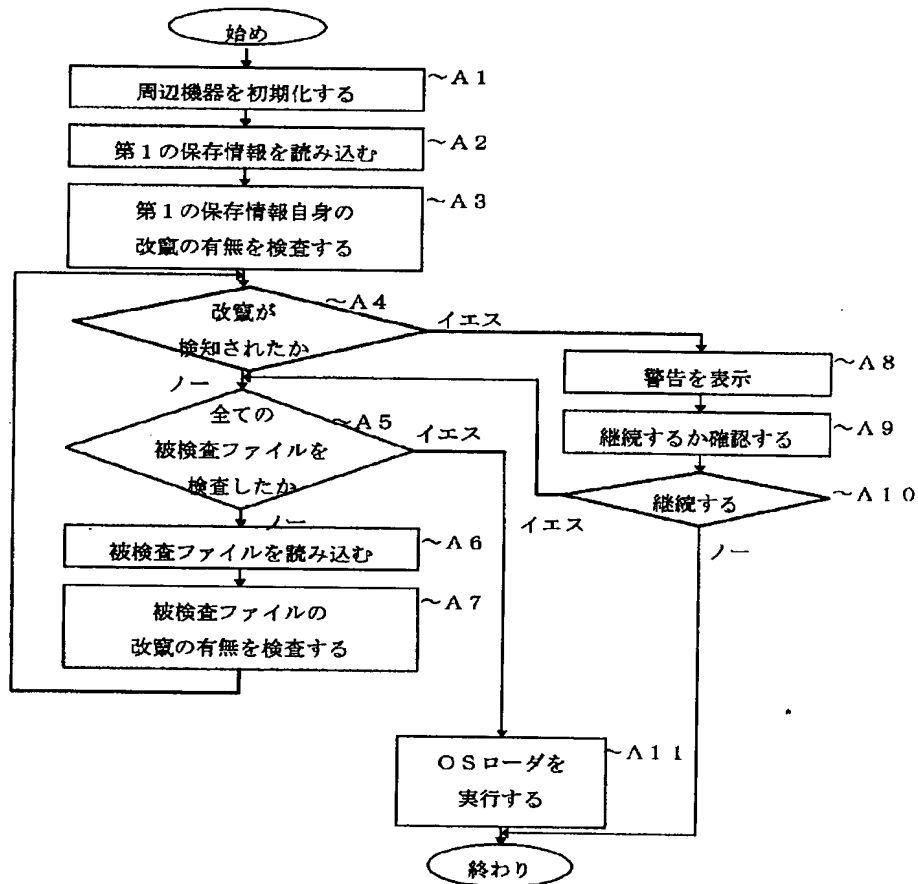
【図1】



【図3】

先頭マーカ	
全体サイズ	
被検査ファイル数 (n)	
第1の被検査ファイル名	第1の被検査ファイルの署名データ
第2の被検査ファイル名	第2の被検査ファイルの署名データ
.....
第nの被検査ファイル名	第nの被検査ファイルの署名データ
第1の保存情報自身の署名データ	

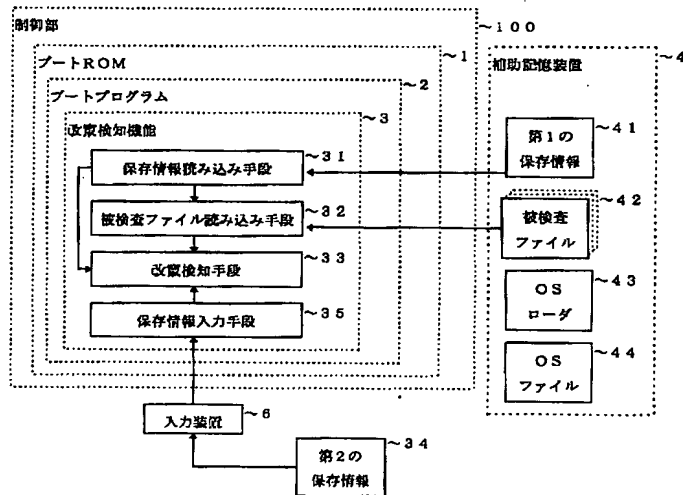
【図2】



【図4】

先頭マーカ	
全体サイズ	
被検査ファイル数 (n)	
第1の被検査ファイル名	第1の被検査ファイルと第2の保存情報とを圧縮した圧縮値
第2の被検査ファイル名	第2の被検査ファイルと第2の保存情報とを圧縮した圧縮値
.....
第nの被検査ファイル名	第nの被検査ファイルと第2の保存情報とを圧縮した圧縮値
第1の保存情報自身の署名データ	

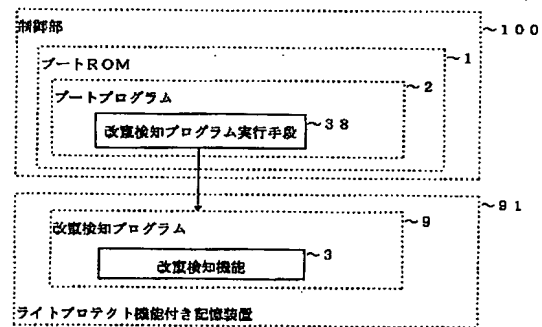
【図5】



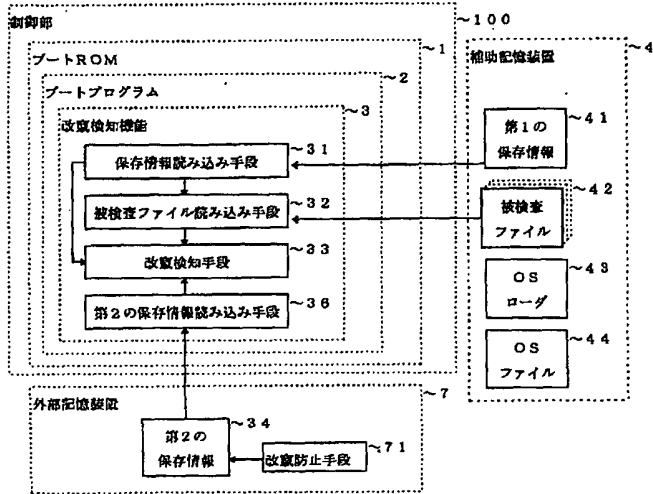
【図6】

先頭マーカ	
全体サイズ	
被検査ファイル数 (n)	
第1の被検査ファイル名	第1の被検査ファイルの圧縮値
第2の被検査ファイル名	第2の被検査ファイルの圧縮値
.....
第nの被検査ファイル名	第nの被検査ファイルの圧縮値
これより上の部分と 第2の保存情報とを 圧縮した圧縮値	

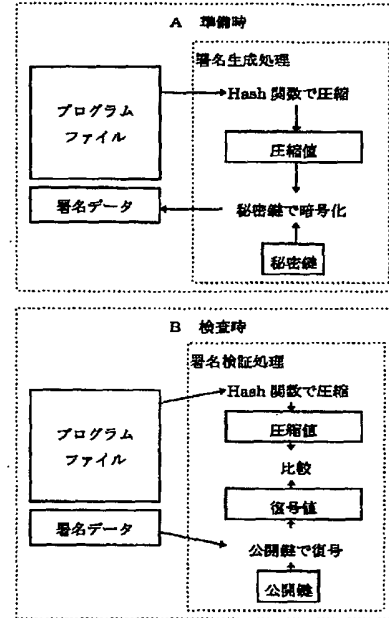
【図9】



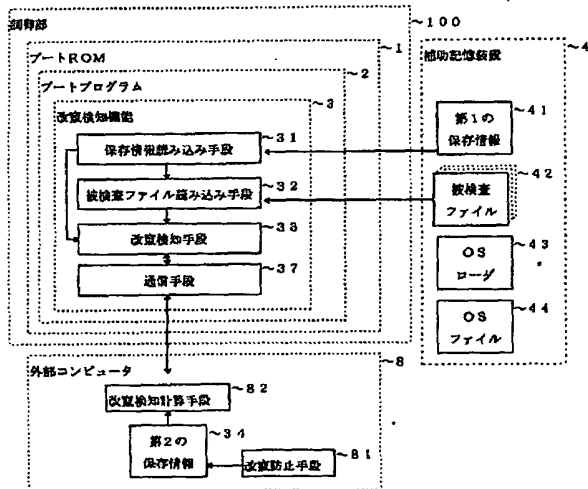
【図7】



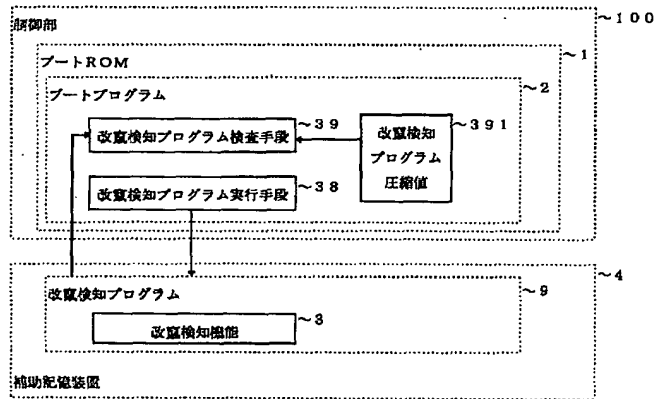
【図12】



【図8】



【図10】



【図11】

